RSAC 2025 Conference

Many Voices.
One Community.

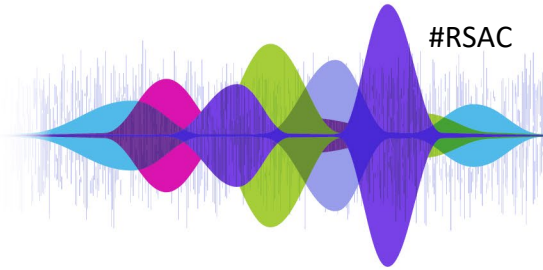
SESSION ID: CRYP-M02A

Comprehensive Robustness Analysis of GCM, CCM, and OCB3

Akiko Inoue[1], **Tetsu Iwata**[2], **Kazuhiko Minematsu**[1,3]

[1] NEC
[2] Nagoya University
[3] Yokohama National University

Disclaimer



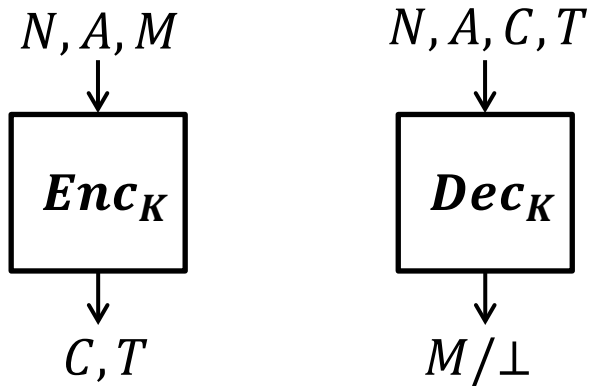
Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference LLC does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

© 2025 RSA Conference LLC or its affiliates. The RSAC and RSAC CONFERENCE logos and other trademarks are proprietary. All rights reserved.

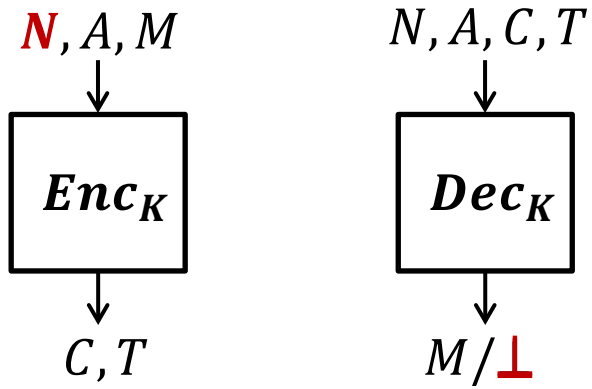
Authenticated encryption; AE

- Symmetric-key crypto to provide privacy (priv) & authenticity (auth)
- Widely deployed in real-world protocols, e.g., TLS, WPA
- Encryption: $Enc_K(N, A, M) = (C, T)$
  - K: key, N: nonce, A: associated data (AD), M: plaintext, C: ciphertext, T: tag
- Decryption: $Dec_K(N, A, C, T) = M$ when inputs are authentic; Otherwise, outputs $\perp$



Misuse of AE

- When using AE, there are some rules we must follow; e.g.,
  - **Nonce respecting**
    .. The user cannot input the same nonce twice in Enc_K.
  - **Decryption respecting**
    .. Dec_K is not allowed to output other than ⊥ if inputs are not authentic.
- If breaking those rules (= **misusing AE**), priv & auth of AE can be broken.
  - e.g., GCM reveals auth key when nonces are misused. [Jou06]



Real-world attack

- Bock et al. investigate the real-world cases in TLS that use **GCM in a nonce-misuse setting**. [BZD+16]
  - Several TLS implementations are vulnerable to forgery attacks.
  - + 70,000 HTTPS servers are also potentially vulnerable due to the use of GCM with **random nonce of short length**.
- Padding oracle attack [Vau02]
  - **Dec leaks info on the decrypted plaintext.**
- Efail attack [PDM+18]
  - Mail server decrypts maliciously manipulated ciphertext and executes the code inside **without verification**.

Nonce-misuse security

- Nonce-Misuse **Resistance** (**NMR**) [RS06]
  - Possible strongest security under nonce misuse;
    No matter how much nonces are misused, priv / auth is maintained.
- Nonce-Misuse **resiLience** (**NML**) [ADL17]
  - Relaxation of NMR; Even if some nonces are misused,
    priv / auth is maintained as long as new nonces are used.
- Tradeoff between security strength and efficiency
  - NMR-secure AE is at most rate-1/2, while NML-secure AE can be rate-1.
  - Rate: measurement of AE efficiency; # processing blocks per 1 E_K call

Decryption-misuse security

- Security under Release of Unverified Plaintext (**RUP**) [ABL+14]
  - Dec function outputs decrypted plaintext **regardless of verification**
  - Priv: plaintext awareness (PA1/PA2)
  - Auth: integrity under RUP (INT-RUP)

Previous work on misuse security of AE

- In the previous AE competitions,
  misuse security is one of the central topics.
  - CAESAR
    - Attack: OCB3 (PA1, INT-RUP), COPA (PA1, INT-RUP)
    - Proof: Minalpher (INT-RUP), AEZ (NMR, RUP), Deoxys-II (NMR)
  - NIST lightweight cryptography
    - Proof: Elephant (NMR-Auth), GIFT-COFB (NML-Priv, NML-Auth),
      Romulus-N (NML-Priv, NML-Auth),
      Romulus-M (NMR-Priv, NMR-Auth, PA1, NMR-INT-RUP)

Research question

- **What about misuse security of the standard AEs?**
  - Vaudenay and Vizár [VV18] show a comprehensive **NMR** comparison of 3rd candidates of CAESAR, plus GCM and CCM.
- There is room for deep analysis, such as **NML** and **RUP**.
- It is important to clarify misuse security of standards since they are *standard* and should be the baseline of AE.
- We pick up GCM, CCM, OCB3, ChaCha20-Poly1305 as standards.

Research question

- **What about misuse security of the standard AEs?**
  - Vaudenay and Vizár [VV18] show a comprehensive **NMR** comparison of 3rd candidates of CAESAR, plus GCM and CCM.
- There is room for deep analysis, such as **NML** and **RUP**.
- It is important to clarify misuse security of standards since they are *standard* and should be the baseline of AE.
- We pick up GCM, CCM, OCB3, ChaCha20-Poly1305 as standards.

NIST standards RFC standards

Contribution

- **Some missing / questionable cases on GCM, CCM, and OCB3**

	RUP		NMR		NML	
	PA1	INT-RUP	Priv	Auth	Priv	Auth
GCM	✗ [ABL+14]	?	✗ [VV18]	✗ [Jou06]	✓* [ADL17]	✗ [ADL17]
CCM	✗ [ABL+14]	?	✗ [VV18]	?	?	?
OCB3	✗ [ABL+14]	✗ [ABL+14]	✗ [Classical]	✗ [VV18]	✘? [ADL17]	✘? [ADL17]
ChaCha20-Poly1305	✗ [ABL+14]	✓ [IMI16, IMI18]	✗ [Classical]	✗ [KDDI17]	✓ [ADL17]	✓ [ADL17]

* GCM is NML-Priv secure only when 96-bit nonce.

Contribution

- **We can prove security in all these missing / questionable cases!**

	RUP		NMR		NML	
	PA1	INT-RUP	Priv	Auth	Priv	Auth
GCM	✗ [ABL+14]	? ⇒ ✔	✗ [VV18]	✗ [Jou06]	✓* [ADL17]	✗ [ADL17]
CCM	✗ [ABL+14]	? ⇒ ✔	✗ [VV18]	? ⇒ ✔	? ⇒ ✔	? ⇒ ✔
OCB3	✗ [ABL+14]	✗ [ABL+14]	✗ [Classical]	✗ [VV18]	✘ ? ⇒ ✔	✘ ? ⇒ ✔†
ChaCha20-Poly1305	✗ [ABL+14]	✓ [IMI16, IMI18]	✗ [Classical]	✗ [KDDI17]	✓ [ADL17]	✓ [ADL17]

* GCM is NML-Priv secure only when 96-bit nonce.
† OCB3 is NML-Auth secure only when AD is empty.

GCM: NIST SP800-38D

- INT-RUP was unknown despite GCM has been extensively studied.
- GCM was considered vulnerable in any misuse scenario.
- Our result shows that this is not the case!

	RUP		NMR		NML	
	PA1	INT-RUP	Priv	Auth	Priv	Auth
GCM	✗ [ABL+14]	? ⇒ ✔	✗ [VV18]	✗ [Jou06]	✓* [ADL17]	✗ [ADL17]
CCM	✗ [ABL+14]	? ⇒ ✔	✗ [VV18]	? ⇒ ✔	? ⇒ ✔	? ⇒ ✔
OCB3	✗ [ABL+14]	✗ [ABL+14]	✗ [Classical]	✗ [VV18]	✘ ? ⇒ ✔	✘ ? ⇒ ✔†
ChaCha20-Poly1305	✗ [ABL+14]	✓ [IMI16, IMI18]	✗ [Classical]	✗ [KDDI17]	✓ [ADL17]	✓ [ADL17]

OCB3: RFC 7253

- Ashur et al. proposed NML-Priv & NML-Auth attacks against **OCB1**.
- They claim, "**We focus on OCB version 1, however our results extend to all versions of OCB.**"

	RUP		NMR		NML	
	PA1	INT-RUP	Priv	Auth	Priv	Auth
GCM	✗ [ABL+14]	? ⇒ ✔	✗ [VV18]	✗ [Jou06]	✓* [ADL17]	✗ [ADL17]
CCM	✗ [ABL+14]	? ⇒ ✔	✗ [VV18]	? ⇒ ✔	? ⇒ ✔	? ⇒ ✔
OCB3	✗ [ABL+14]	✗ [ABL+14]	✗ [Classical]	✗ [VV18]	✘ ? ⇒ ✔	✘ ? ⇒ ✔†
ChaCha20-Poly1305	✗ [ABL+14]	✓ [IMI16, IMI18]	✗ [Classical]	✗ [KDDI17]	✓ [ADL17]	✓ [ADL17]

OCB3: RFC 7253

- We show their attacks are **not** applicable to OCB3, and we can prove both security.
- Note: AD needs to be empty for NML-Auth.

	RUP		NMR		NML	
	PA1	INT-RUP	Priv	Auth	Priv	Auth
GCM	✗ [ABL+14]	? ⇒ ✔	✗ [VV18]	✗ [Jou06]	✓* [ADL17]	✗ [ADL17]
CCM	✗ [ABL+14]	? ⇒ ✔	✗ [VV18]	? ⇒ ✔	? ⇒ ✔	? ⇒ ✔
OCB3	✗ [ABL+14]	✗ [ABL+14]	✗ [Classical]	✗ [VV18]	✘ ? ⇒ ✔	✘ ? ⇒ ✔†
ChaCha20-Poly1305	✗ [ABL+14]	✓ [IMI16, IMI18]	✗ [Classical]	✗ [KDDI17]	✓ [ADL17]	✓ [ADL17]

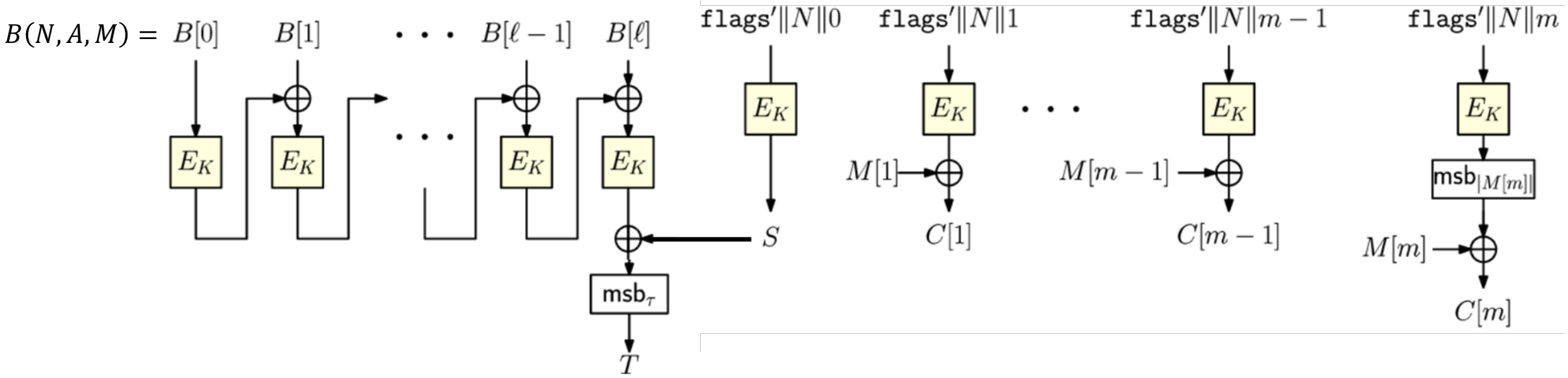
CCM: NIST SP 800-38C

- Only few works on provable security exist.
- Previous research only revealed trivial cases.
- In this talk, we give details of these results.

	RUP		NMR		NML	
	PA1	INT-RUP	Priv	Auth	Priv	Auth
GCM	✗ [ABL+14]	? ⇒ ✔	✗ [VV18]	✗ [Jou06]	✓* [ADL17]	✗ [ADL17]
CCM	✗ [ABL+14]	? ⇒ ✔	✗ [VV18]	? ⇒ ✔	? ⇒ ✔	? ⇒ ✔
OCB3	✗ [ABL+14]	✗ [ABL+14]	✗ [Classical]	✗ [VV18]	✘ ? ⇒ ✔	✘ ? ⇒ ✔†
ChaCha20-Poly1305	✗ [ABL+14]	✓ [IMI16, IMI18]	✗ [Classical]	✗ [KDDI17]	✓ [ADL17]	✓ [ADL17]

CCM

- MAC-then-Enc: CBC-MAC + CTR
  - CBC-MAC output is XORed with encryption of nonce & zero.
  - Input of CBC-MAC: encoded string $B(N, A, M) = B[0] \,||\, B[1] \,||\, \cdots \,||\, B[\ell]$
  - $B(N, A, M)$ is **prefix-free**:
    there is no $B(\cdot,\cdot,\cdot)$ corresponding to the prefix of other $B(\cdot,\cdot,\cdot)$.

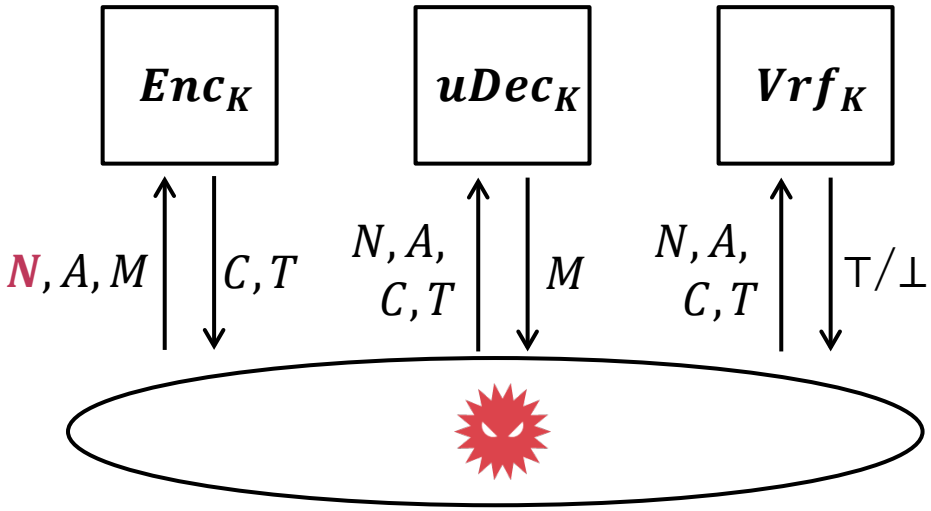


CCM

- MAC-then-Enc: CBC-MAC + CTR
  - CBC-MAC output is XORed with encryption of nonce & zero.
  - Input of CBC-MAC: encoded string $B(N, A, M) = B[0] \,||\, B[1] \,||\, \cdots \,||\, B[\ell]$
  - $B(N, A, M)$ is **prefix-free**:
    there is no $B(\cdot,\cdot,\cdot)$ corresponding to the prefix of other $B(\cdot,\cdot,\cdot)$.
- Existing misuse security analysis
  - Fouque et al. [FMVZ08], Vaudenay and Vizár [VV18]: NMR-Auth attack
    - Comp: $O\left(2^{n/2}\right)$, where n is block size
  - Other than these results, as far as we investigated,
    no results on misuse security of CCM.

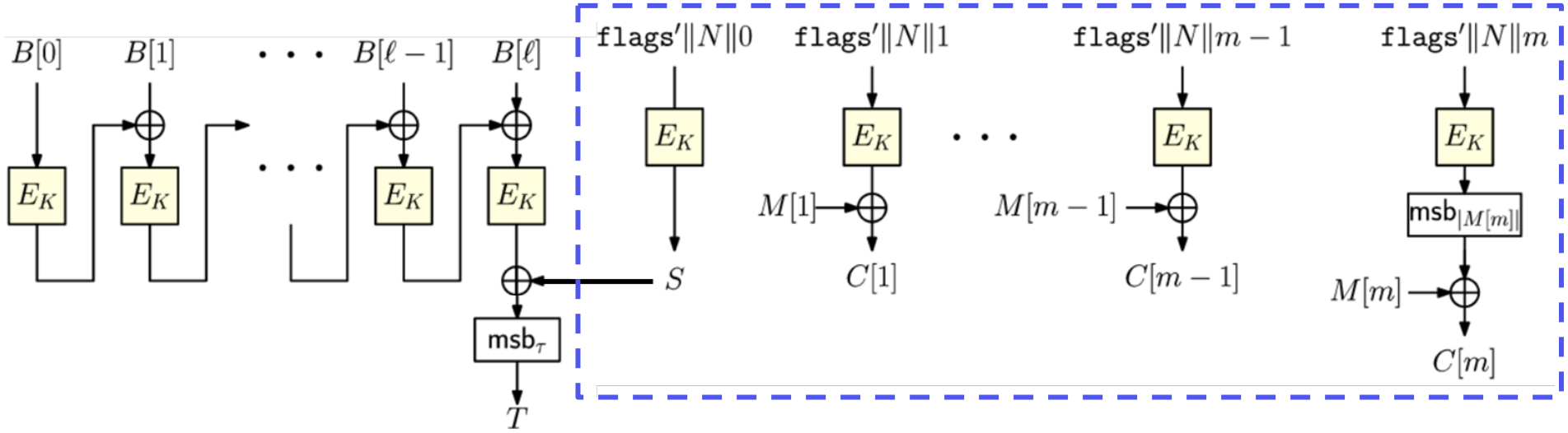
Misuse auth on CCM

- **We show CCM has NMR-INT-RUP security.**
  - NMR-INT-RUP implies all missing misuse auth on CCM: INT-RUP, NMR-Auth, NML-Auth.
- Separated syntax
  - $Enc_K(N, A, M) = (C, T)$,
  - $uDec_K(N, A, C, T) = \boldsymbol{M}$,
  - $Vrf_K(N, A, C, T) = \top/\bot$
- NMR-INT-RUP
  - Prob. of forging under Enc, $uDec$, Vrf queries
  - Nonce can be repeated even in Enc_K



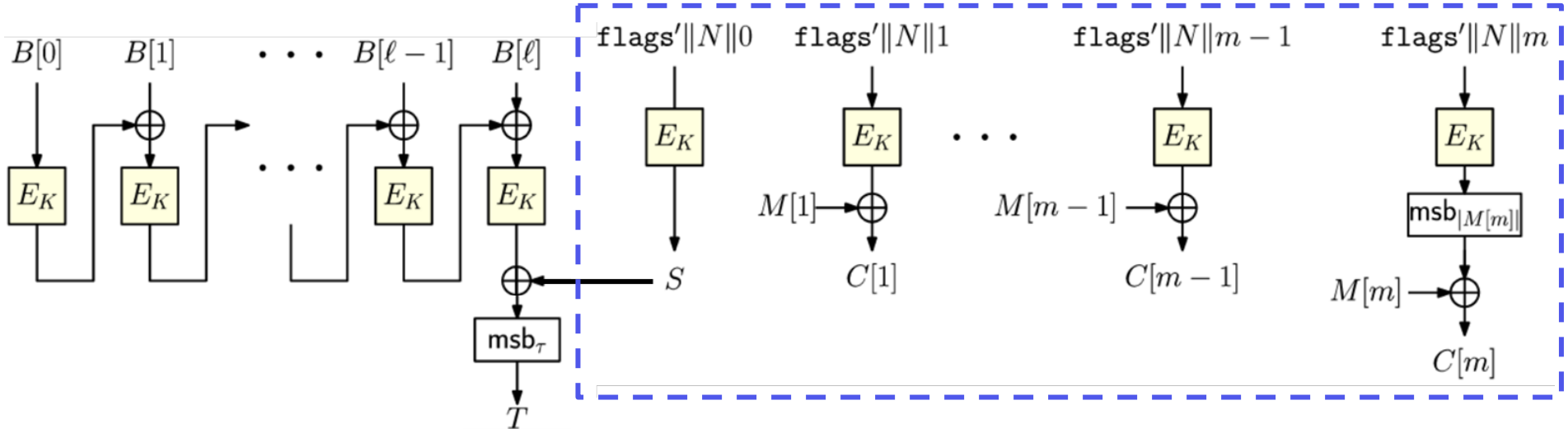
Proof overview

- Reduce NMR-INT-RUP of CCM to a variant of UF-CMA of CBC-MAC.
  - UF-CMA: unforgeability under chosen message attacks
  - Define UF-CMA+: UF-CMA with additional accesses to [ ].
- CBC-MAC is secure if the input is prefix-free [PR97].
  - Recall: CBC-MAC input in CCM is prefix-free; this holds even if nonce misuse.
  - **We can prove security of CBC-MAC under UF-CMA+ as well!**



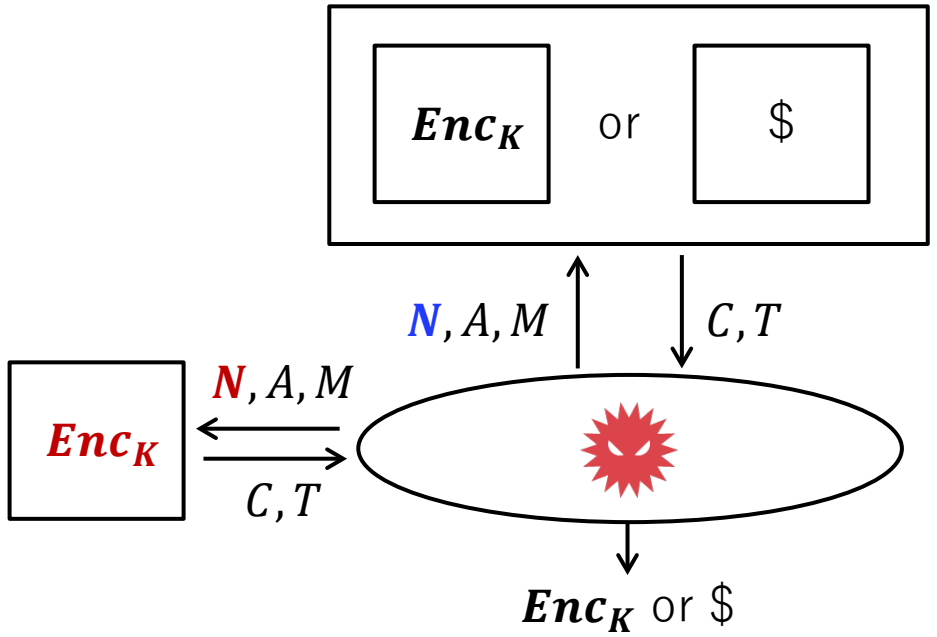
Security bound

- Prove UF-CMA+ of CBC-MAC by H-coefficient.
  - Bad events: collision at E_K input
- We show that CCM has NMR-INT-RUP security of $\boldsymbol{O\left(\min\left\{2^{n/2}, 2^{\tau}\right\}\right)}$
  - n: block size, τ: tag length
  - **Same security level as standard auth**



Misuse priv on CCM

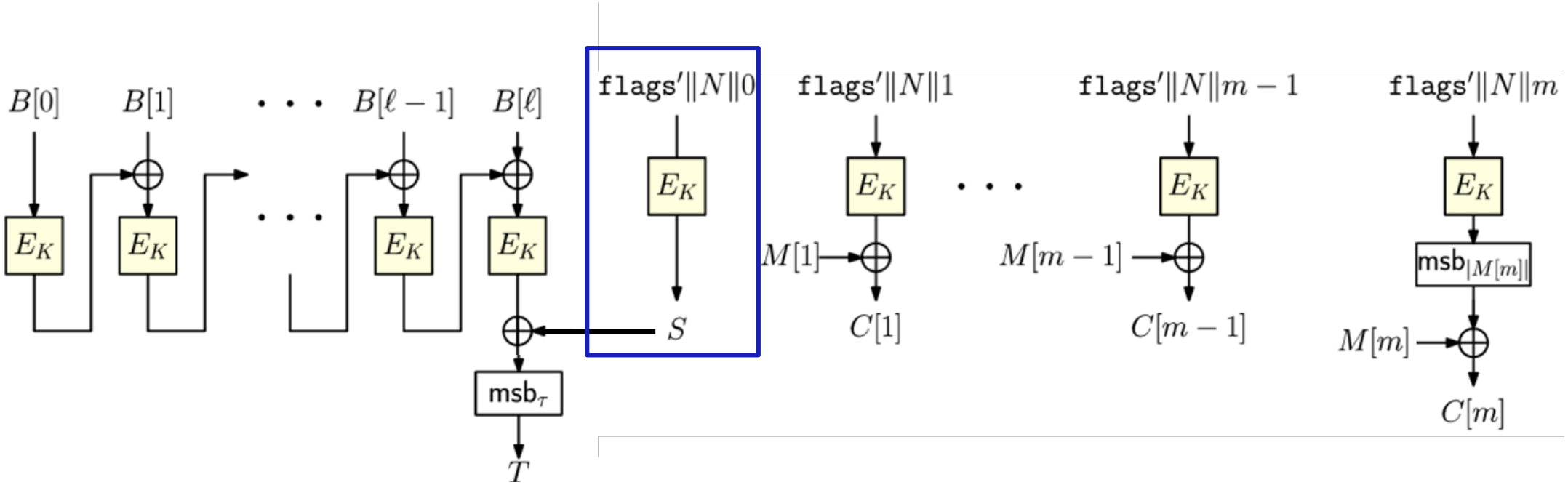
- CCM has NMR-Auth, but does not have NMR-Priv.
  - Encryption of CCM is CTR
  - Same nonces produce same key streams.
- What about NML-Priv?
  - Def: Prob. of distinguishing $\boldsymbol{Enc_K}$ from \$ with **nonce-respecting queries** under **nonce-repeating queries** to $\boldsymbol{Enc_K}$



- We can prove CCM has NML-Priv of $\boldsymbol{O(2^{n/2})}$.
  - Even if some nonces are misused, the adv. cannot violate the security of the keystream derived from other non-misused nonces.

Optimization of CCM: CCM2

- CCM2 omits the mask generation for the tag (▭) in CCM.
- CCM2 reduces 1 E_k call &
  achieves the same misuse security as CCM.
  - NMR-INT-RUP & NML-Priv



Performance of CCM2: short-input case

- Adomnicăi et al. [AMS23] show that general-purpose AE require at least 3 E_K calls to process non-empty input.
- CCM2 achieves this minimum number, which implies excellent performance for very short inputs, e.g., on microcontrollers.

	GCM*	CCM	OCB3	COFB	**CCM2**
E_K Calls	3 + 2 mul	4	4	3	**3**
Parallel-lizability	3 + 1 mul + 1 mul	2+2	2+2	1+1+1	**2+1**

* When using 96-bit nonce

Lessons learned

- When using AE, please check its rules and follow them.
- Using misuse-resistant AE is good choice … but,
- There is a tradeoff between security strength and efficiency.
- It is important to select AE that are suitable to your environment, & check what happens if the rules of these AE are violated.
  - If you consider using GCM, CCM, OCB3, or ChaCha20-Poly1305, please check the summary table in our research 😉

Summary

- Provided a comprehensive overview on the misuse security of the following four standard AEs.
  - CCM & CCM2 have the strongest misuse strength among the four AEs.

	RUP		NMR		NML	
	PA1	INT-RUP	Priv	Auth	Priv	Auth
GCM	✗ [ABL+14]	? ⇒ ✔	✗ [VV18]	✗ [Jou06]	✓* [ADL17]	✗ [ADL17]
CCM/CCM2	✗ [ABL+14]	? ⇒ ✔	✗ [VV18]	? ⇒ ✔	? ⇒ ✔	? ⇒ ✔
OCB3	✗ [ABL+14]	✗ [ABL+14]	✗ [Classical]	✗ [VV18]	✘ ? ⇒ ✔	✘ ? ⇒ ✔†
ChaCha20-Poly1305	✗ [ABL+14]	✓ [IMI16, IMI18]	✗ [Classical]	✗ [II16]	✓ [ADL17]	✓ [ADL17]

RSAC™ 2025 Conference

Reference & Appendix

Reference

- [Jou06]: Joux, A.: Authentication Failures in NIST version of GCM. Public comments to NIST (2006), http://csrc.nist.gov/groups/ST/toolkit/BCM/comments.html
- [BZD+16]: Böck, H., Zauner, A., Devlin, S., Somorovsky, J., Jovanovic, P.: Nonce-Disrespecting Adversaries: Practical Forgery Attacks on GCM in TLS. In: WOOT. USENIX Association (2016)
- [Vau02]: Vaudenay, S.: Security flaws induced by CBC padding - applications to SSL, IPSEC, WTLS... In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332,pp. 534–546. Springer, Heidelberg (Apr / May 2002). https://doi.org/10.1007/3-540-46035-7_35
- [PDM+18]: Poddebniak, D., Dresen, C., Müller, J., Ising, F., Schinzel, S., Friedberger, S., Somorovsky, J., Schwenk, J.: Efail: Breaking S/MIME and OpenPGP email encryption using exfiltration channels. In: Enck, W., Felt, A.P. (eds.) USENIX Security 2018. pp. 549–566. USENIX Association (Aug 2018)
- [RS06]: Rogaway, P., Shrimpton, T.: A provable-security treatment of the key-wrap problem. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 373–390. Springer, Heidelberg (May / Jun 2006). https://doi.org/10.1007/11761679_23

Reference

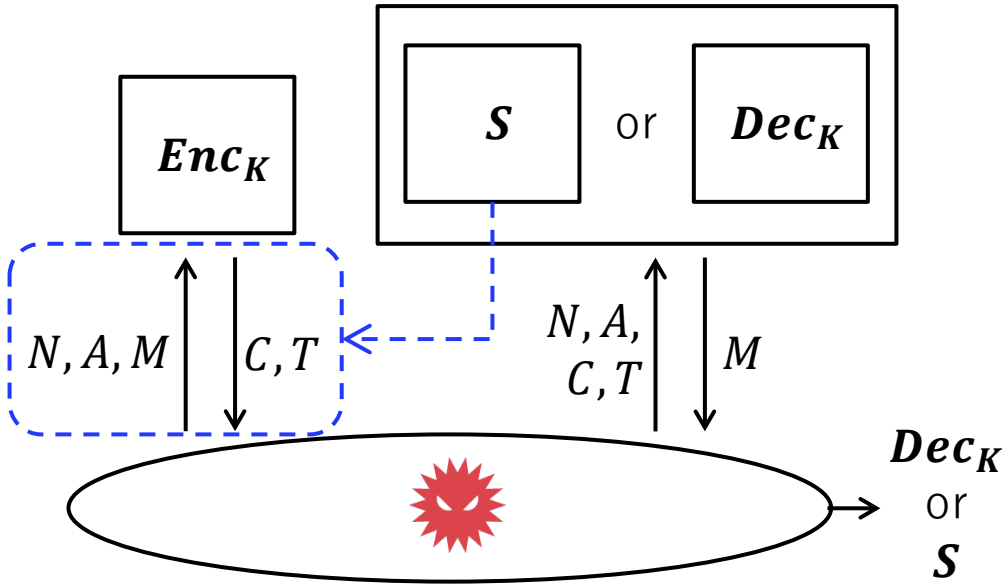
- [ADL17]: Ashur, T., Dunkelman, O., Luykx, A.: Boosting authenticated encryption robustness with minimal modifications. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017,Part III. LNCS, vol. 10403, pp. 3–33. Springer, Heidelberg (Aug 2017). https://doi.org/10.1007/978-3-319-63697-9_1
- [ABL+14]: Andreeva, E., Bogdanov, A., Luykx, A., Mennink, B., Mouha, N., Yasuda, K. (2014). How to Securely Release Unverified Plaintext in Authenticated Encryption. In: Sarkar, P., Iwata, T. (eds) Advances in Cryptology – ASIACRYPT 2014. ASIACRYPT 2014. Lecture Notes in Computer Science, vol 8873.
- [VV18]: Vaudenay, S., Vizár, D.: Can caesar beat galois? - Robustness of CAESAR candidates against nonce reusing and high data complexity attacks. In: Preneel, B., Vercauteren, F. (eds.) ACNS 18. LNCS, vol. 10892, pp. 476–494. Springer, Heidelberg (Jul 2018). https://doi.org/10.1007/978-3-319-93387-0_25
- [IMI16]: Imamura, K., Minematsu, K., Iwata, T.: Integrity analysis of authenticated encryption based on stream ciphers. In: Chen, L., Han, J. (eds.) ProvSec 2016. LNCS, vol. 10005, pp. 257–276. Springer, Heidelberg (Nov 2016). https://doi.org/10.1007/978-3-319-47422-9_15
- [IMI18]: Imamura, K., Minematsu, K., Iwata, T.: Integrity analysis of authenticated encryption based on stream ciphers. Int. J. Inf. Sec. 17(5), 493–511 (2018)

Reference

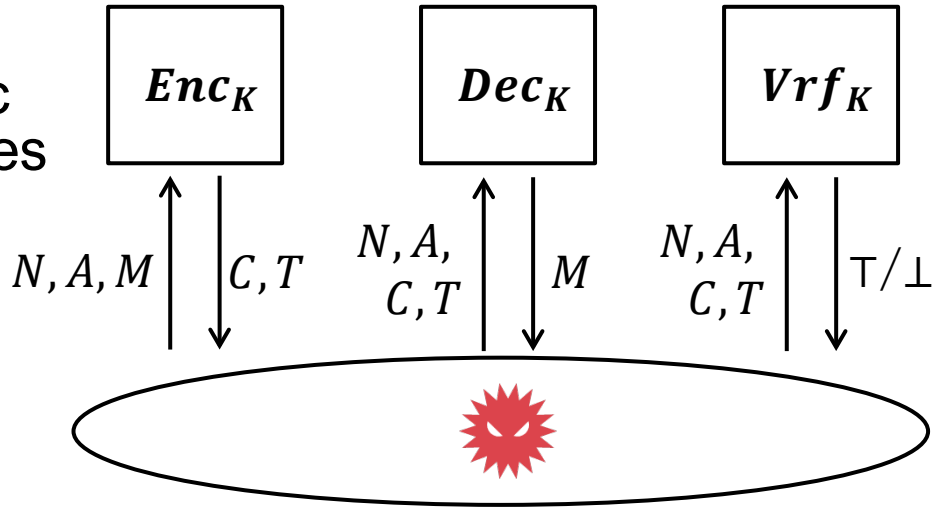
- [KDDI17]: KDDI Research, Inc.: Security Analysis of ChaCha20-Poly1305 AEAD. CRYPTREC-EX-2601-2016 (2017)
- [FMVZ08]: Fouque, P.A., Martinet, G., Valette, F., Zimmer, S.: On the security of the CCM encryption mode and of a slight variant. In: Bellovin, S.M., Gennaro, R., Keromytis, A.D., Yung, M. (eds.) ACNS 08. LNCS, vol. 5037, pp. 411–428. Springer, Heidelberg (Jun 2008). https://doi.org/10.1007/978-3-540-68914-0_25
- [PR97]: Petrank, E., Rackoff, C.: CBC MAC for real-time data sources. Journal of Cryptology 13(3), 315–338 (Jun 2000). https://doi.org/10.1007/s001450010009
- [AMS23]: Adomnicai, A., Minematsu, K., Shikata, J.: Authenticated encryption for very short inputs. In: CT-RSA 2023. pp. 553–572. LNCS, Springer, Heidelberg (Feb 2023). https://doi.org/10.1007/978-3-031-30872-7_21

RUP: Definition

- Separated syntax:
  - $Enc_K(N, A, M) = (C, T)$,
  - $Dec_K(N, A, C, T) = M$,
  - $Vrf_K(N, A, C, T) = \top/\bot$



- Privacy: Plaintext awareness (PA1/2)
  - Prob. of distinguishing a simulator mimicking Dec without a key from genuine Dec under Enc queries

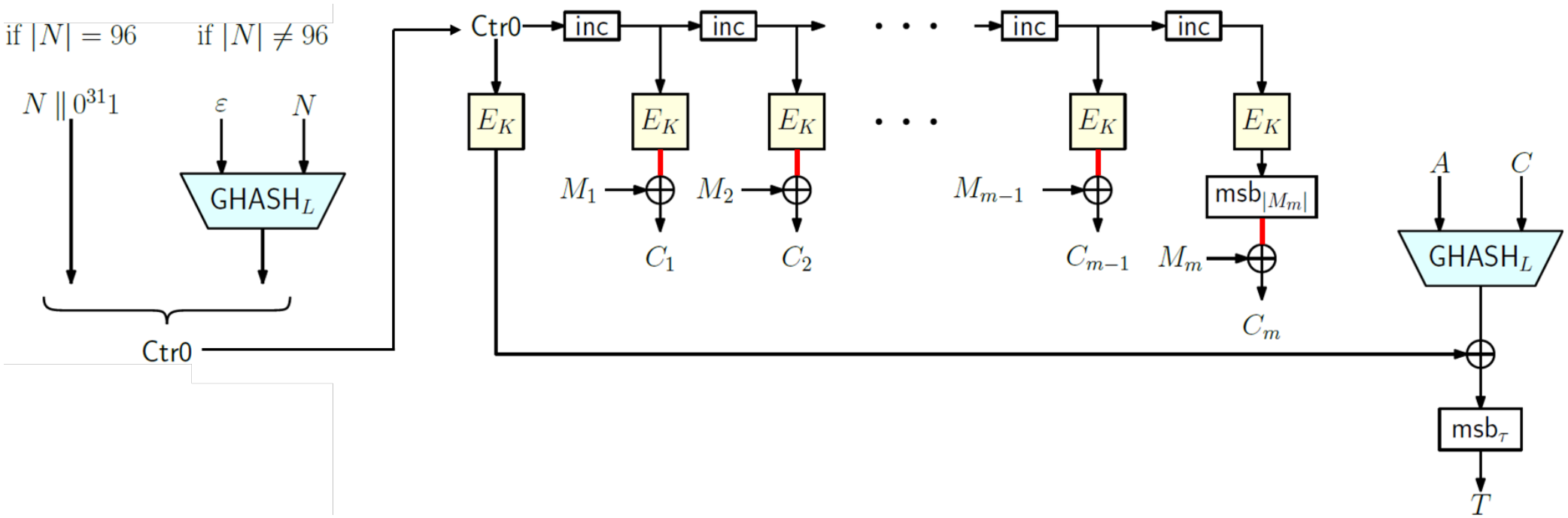


- Authenticity: Integrity under RUP (INT-RUP)
  - Prob. of forging under Enc, Dec, Vrf queries

RSAC 2025 Conference

GCM INT-RUP proof: First observation

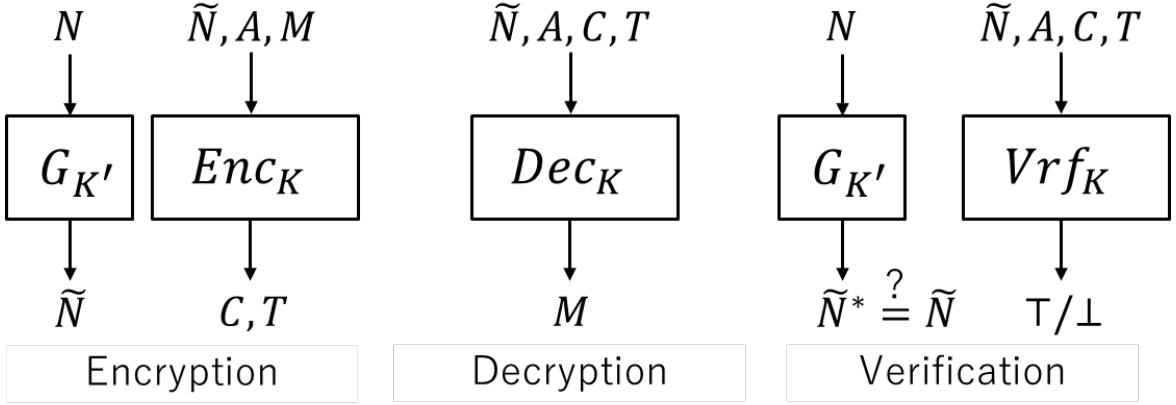
- Recall: the adversary can access to $Dec_K(N, A, C, T) = M$
  - Dec reveals only the **key stream** on the queried nonce.
  - If querying the same nonce as that in Enc queries, adv. can only obtain key stream and it cannot mount Joux's forbidden attack [Joux06].



Application of our result

- GCM is **not** PA1 secure. Does having “only” INT-RUP security make sense?
- One answer: Nonce decoy [ABL+14]
  - It converts a scheme into a PA1 one, **if the baseline scheme is PA1 under random IV.**
  - It encrypts an input nonce and outputs its value in addition to typical AE outputs.
  - (Recall) CTR is PA1 under random IV → **CTR (GCM) fulfills this requirement.**
  - **Nonce decoy preserves INT-RUP** of the baseline scheme.

→ **GCM can be converted to be RUP secure just applying nonce decoy!**



Enc_K, Dec_K, Vrf_K : Baseline AE

$G_{K'}$: PRF s.t. $G_{K'}: \{0,1\}^{|N|} \to \{0,1\}^{|N|}$

Output of enc: $(N, \widetilde{N}, A, C, T)$

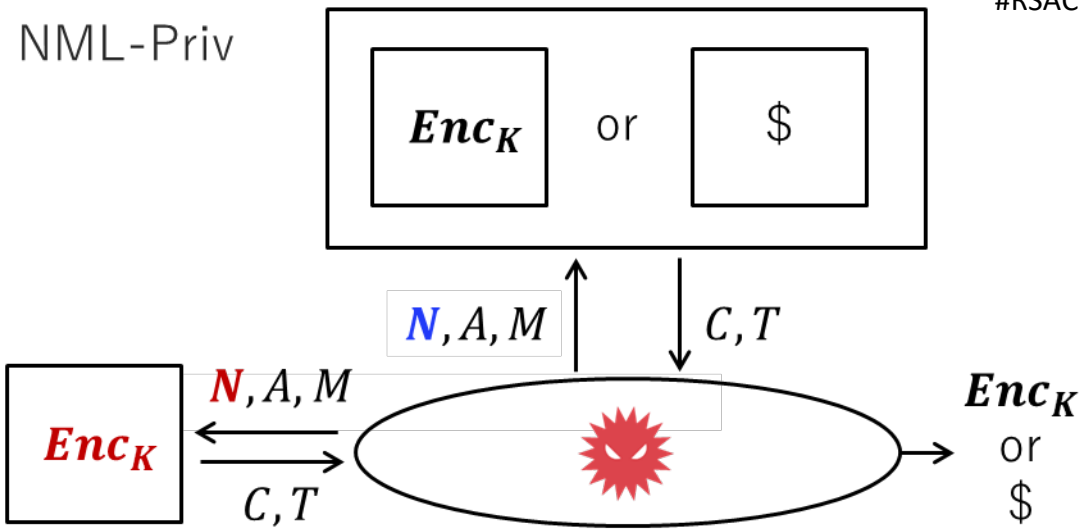
Comparison

- PRF-to-IV and GCM-RUP do not rely on INT-RUP security of GCM
- **GCM + Nonce decoy works well because GCM is INT-RUP**
  - 👍 RUP security, black box, comparable increase of computation, and low dec latency
  - 👎 Ciphertext expansion for an encrypted nonce

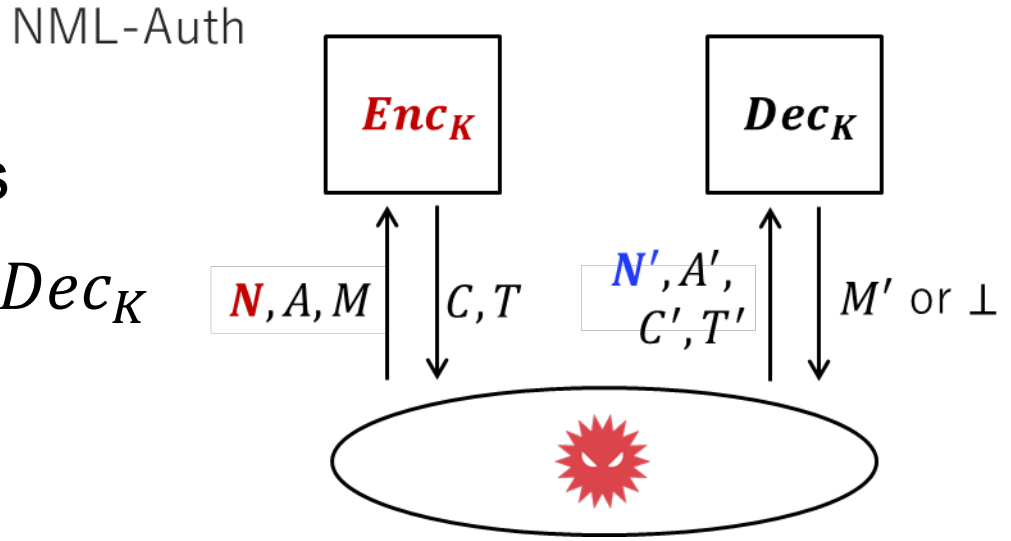
Scheme	Security: PA1 + INT-RUP	Black box conversion	Increase of computation	Dec latency (until dec part starts)	Ciphertext expansion
GCM + PRF-to-IV	○	○	PRF(N,A,M)	None	|N|
GCM-RUP	○	×	≤ 1 BC call	GHASH(A, C) + GHASH(N, ε)	None
GCM + Nonce decoy	○	○	1 BC call	None	|N|

NML-Auth

- Syntax:
  - $Enc_K(N, A, M) = (C, T)$,
  - $Dec_K(N, A, C, T) = M/\perp$

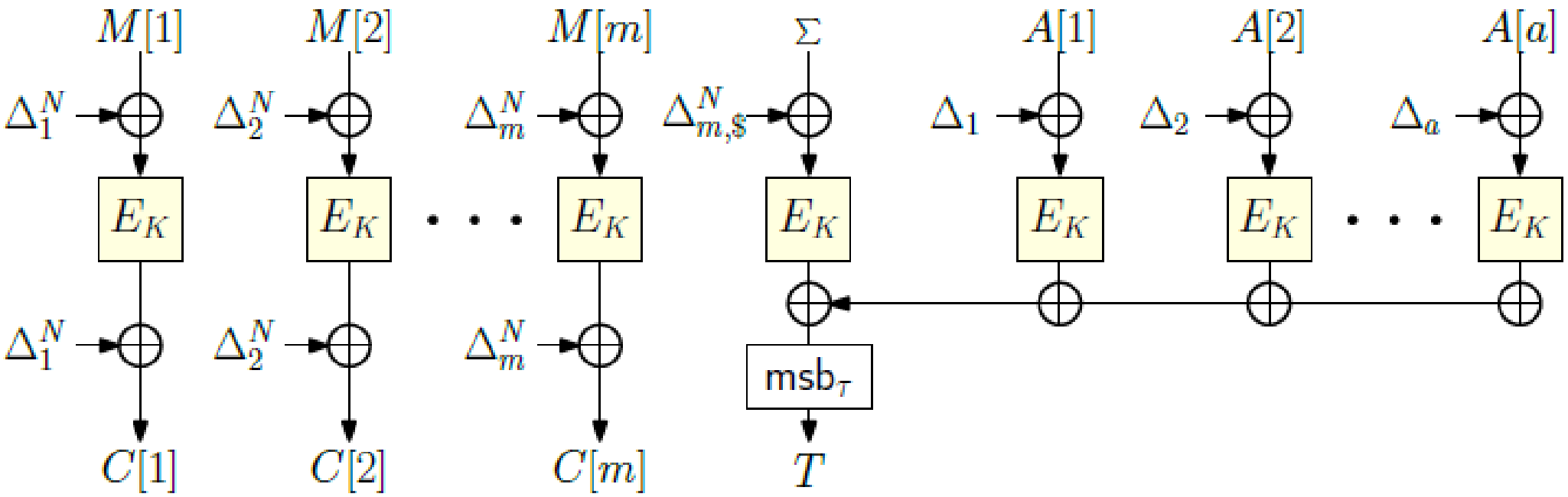


- NML-Auth
  - Prob. of forging under Enc, Dec queries
  - Nonce can be repeated both Enc_K and Dec_K
  - In Dec_K, adv. cannot use nonces which is repeated in Enc_K



RSAC 2025 Conference

OCB3


$M[1]$
$M[2]$
$M[m]$
Σ
$A[1]$
$A[2]$
$A[a]$
Δ_1^N
Δ_2^N
Δ_m^N
$\Delta_{m,\$}^N$
Δ_1
Δ_2
Δ_a
E_K
msb_τ
$C[1]$
$C[2]$
$C[m]$
T

OCB1

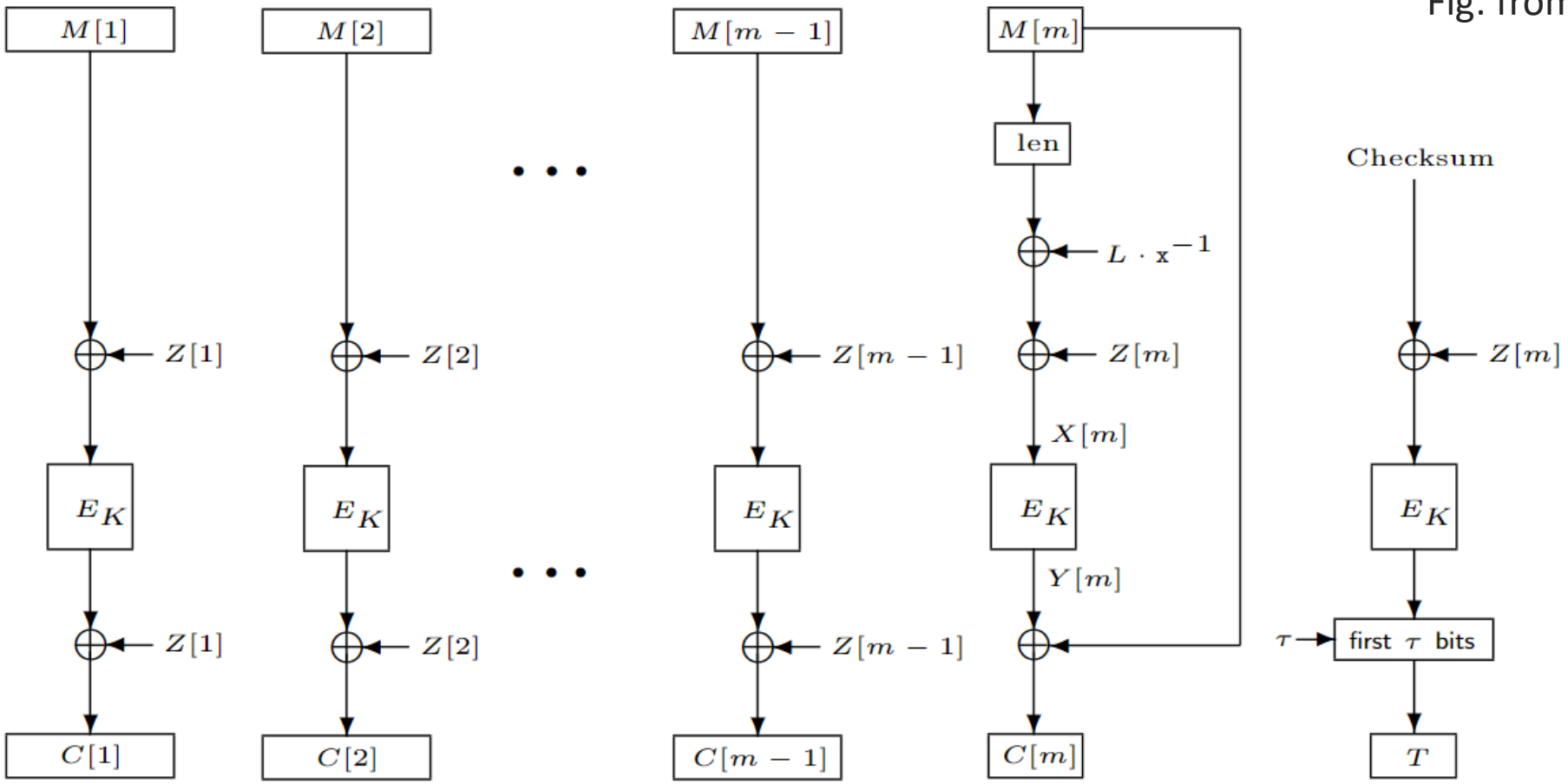
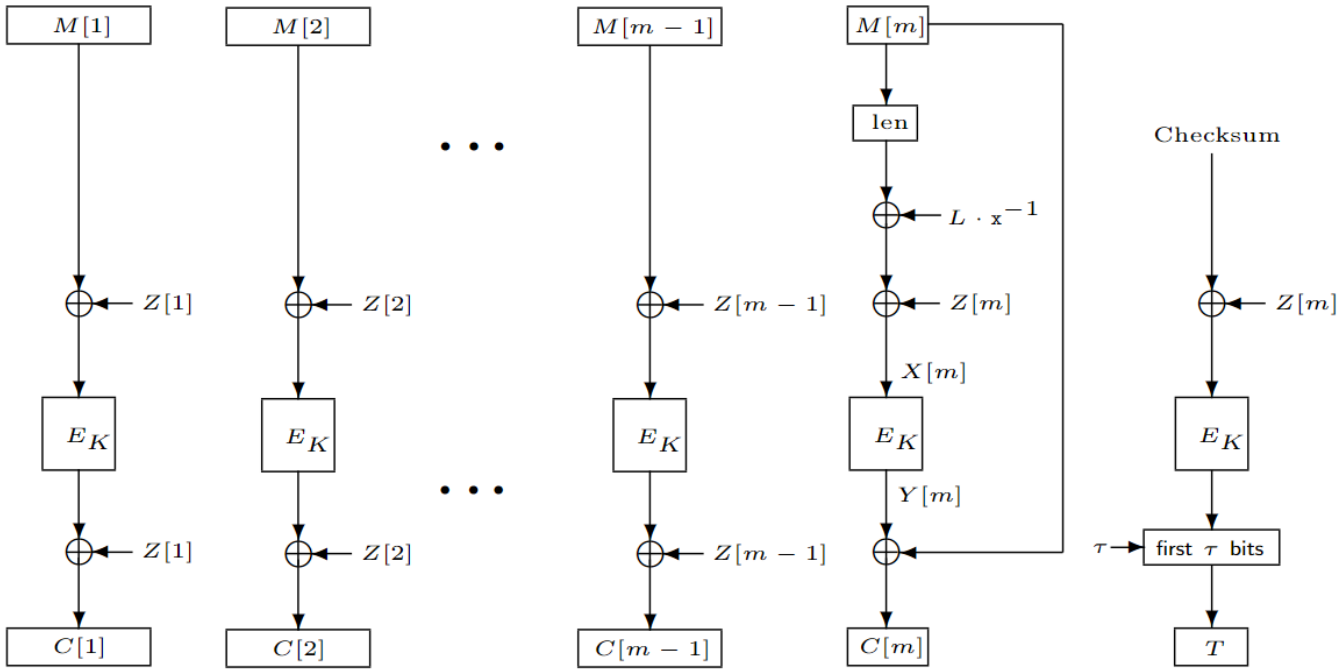


Fig: from [RBBK01]

RSAC 2025 Conference

Overview of NML attacks on OCB1 [ADL17]

Fig: from [RBBK01]



- Spec. of OCB1
  - Masks are derived from two seeds.
    - $L = E_K(0^n), R = E_K(L \oplus N)$
  - OCB1 does **not** use a special mask for tag derivation XE.
    - We can use $Z[m]$ for both tag derivation and message processing.
- Attack Idea
  - Nonce-misuse queries with various lengths allow the adv. to query XE and XEX with the same nonce and index.
  - Adv. can recover the mask value, and by repeating this, it also can recover two mask seeds.
  - Once mask seeds are recovered, adv. can mount privacy and authenticity attacks against nonce-respecting queries.

NML analysis on OCB3

- We cannot apply the previous OCB1 attack to OCB3.
  - In OCB3, tag derivation XE uses a special mask, unlike OCB1.
  - Various length queries do not reveal the mask.
- We can prove NML-Priv.
  - Recall: security of OCB3: ΘCB sec. + XEX* sec.
  - XEX* sec.: no restriction on nonce repetition
    ⇒ It is enough to prove the NML-Priv of ΘCB.
  - We can prove it in the similar manner as the standard security proof of ΘCB.
- We can prove NML-Auth when AD is empty in the same manner as NML-Priv.
  - Why AD has to be empty?: AD processing is **not** nonce-based.
    ⇒ Two NM queries can reveal diff. of AD processing value as shown in [VV18].

RSAC 2025 Conference

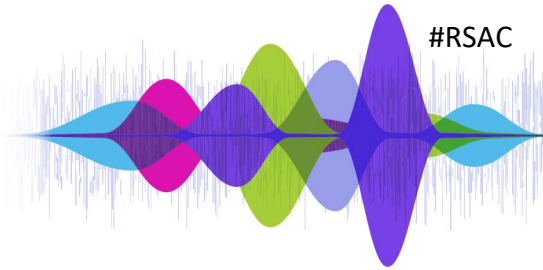
Many Voices.
One Community.

SESSION ID: CRYP-M02B

Meet-in-the-Middle Attack on Primitives with Binary Matrix Linear Layer

Kuntong Li

Masters Student
Shandong University



Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference LLC does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

© 2025 RSA Conference LLC or its affiliates. The RSAC and RSAC CONFERENCE logos and other trademarks are proprietary. All rights reserved.

Outline

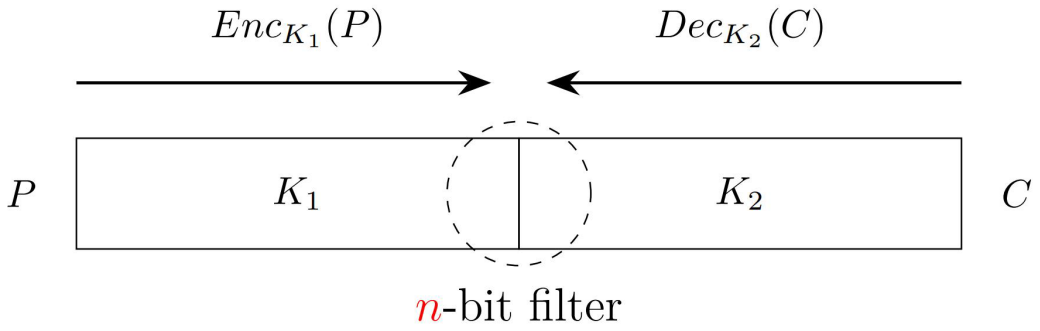
- Background and preliminary
- New models for linear layer with binary matrix
- Key recovery attack on Midori64
- Preimage attack on weakened Camellia and Aria
- Conclusion

RSAC 2025 Conference

Background and preliminary

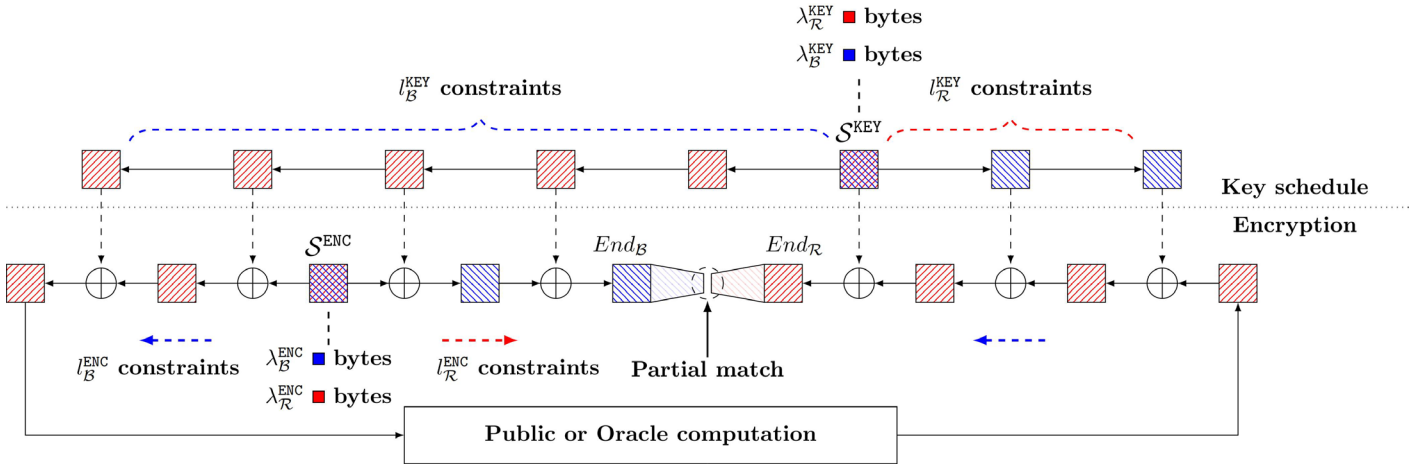
Meet-in-the-Middle(MitM) attack

Meet-in-the-Middle(MitM) attack was first introduced by Diffie and Hellman[DH77] in 1977.



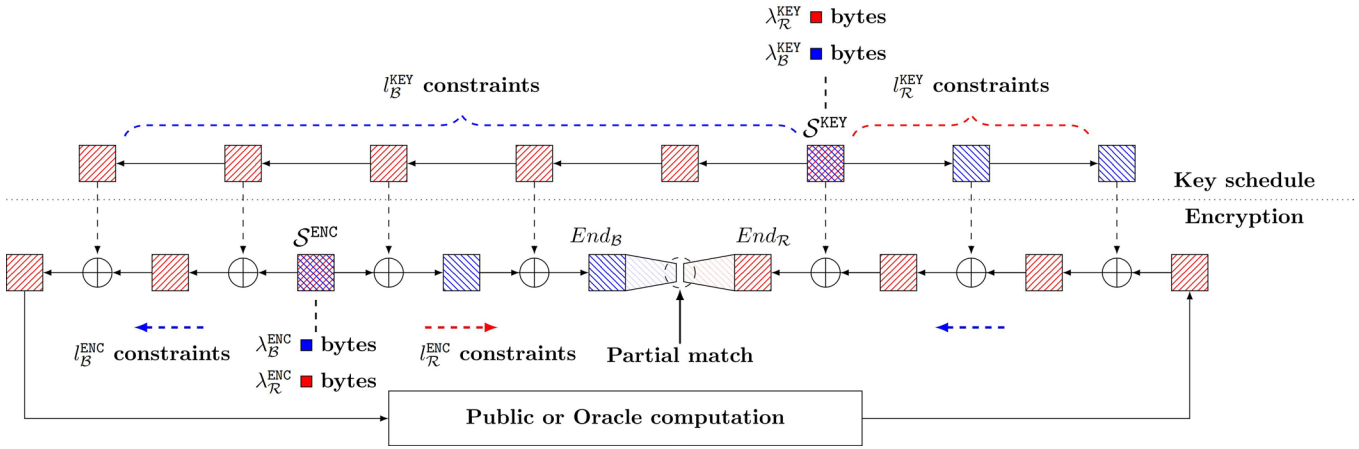
- Forward Encryption: store K_1 in table $L_1[Enc_{K_1}(P)]$.
- Backward Decryption: store K_2 in table $L_2[Dec_{K_2}(C)]$.
- Find match between L_1 and L_2.
- $2^{|K_1|+|K_2|-n} (< 2^{|K_1|+|K_2|})$ keys survived.

Framework of the Meet-in-the-Middle Attack



- MILP-based preimage attack [EC:BDG+21]
- Key-recovery and collision attacks, Table-based methods [C:DHSLWH21]
- Superposition (SupP) states, Bi-directional attribute propagation (BiDir) [C:BGST22]

Framework of the Meet-in-the-Middle Attack

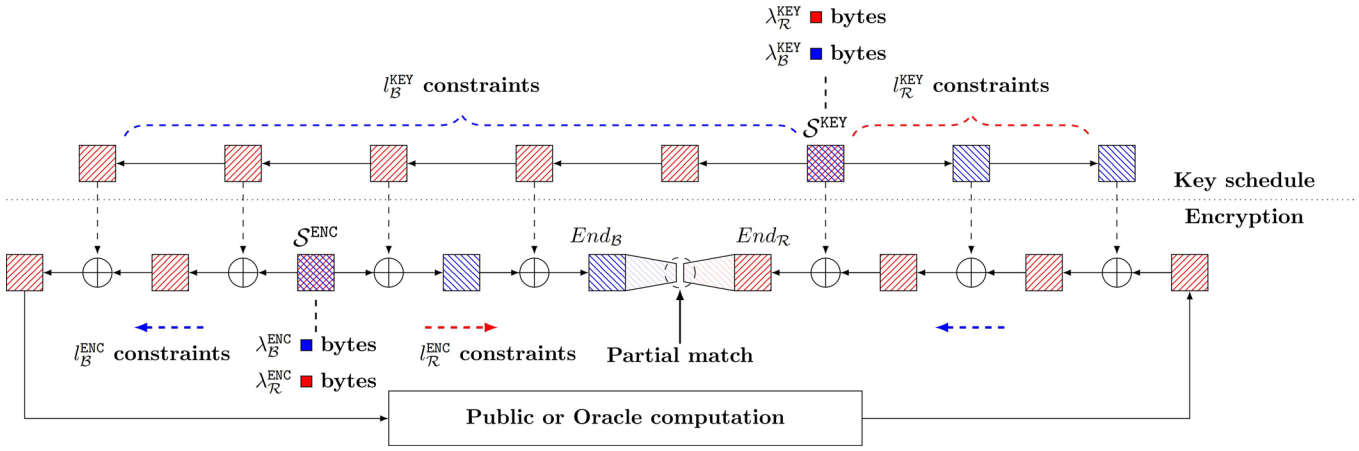


- Initial structure:

$$\begin{cases} \text{Starting states: } \mathcal{S}^{\mathtt{ENC}}, \mathcal{S}^{\mathtt{KEY}} \\ \lambda_{\mathcal{B}} \blacksquare,\ \lambda_{\mathcal{R}} \blacksquare,\ \lambda_{\mathcal{G}} \blacksquare \text{ in starting states.} \\ l_{\mathcal{B}} \text{ constraints on } \blacksquare,\ l_{\mathcal{R}} \text{ constraints on } \blacksquare. \end{cases}$$

- Forward: $\mathrm{DoF}_{\mathcal{B}} = \lambda_{\mathcal{B}} - l_{\mathcal{B}}$ values of $\blacksquare$.
- Backward: $\mathrm{DoF}_{\mathcal{R}} = \lambda_{\mathcal{R}} - l_{\mathcal{R}}$ values of $\blacksquare$.
- Match: DoM bytes from $End_{\mathcal{B}} = End_{\mathcal{R}}$.

Framework of the Meet-in-the-Middle Attack



- Initial structure:

$$\begin{cases} \text{Starting states: } \mathcal{S}^{\text{ENC}}, \mathcal{S}^{\text{KEY}} \\ \lambda_{\mathcal{B}} \blacksquare,\ \lambda_{\mathcal{R}} \blacksquare,\ \lambda_{\mathcal{G}} \blacksquare \text{ in starting states.} \\ l_{\mathcal{B}} \text{ constraints on } \blacksquare,\ l_{\mathcal{R}} \text{ constraints on } \blacksquare. \end{cases}$$

- Forward: $\text{DoF}_{\mathcal{B}} = \lambda_{\mathcal{B}} - l_{\mathcal{B}}$ values of $\blacksquare$.
- Backward: $\text{DoF}_{\mathcal{R}} = \lambda_{\mathcal{R}} - l_{\mathcal{R}}$ values of $\blacksquare$.
- Match: DoM bytes from $End_{\mathcal{B}} = End_{\mathcal{R}}$.

Attack Framework

1. Choose constants in $\mathcal{S}^{\text{ENC}}$ and $\mathcal{S}^{\text{KEY}}$ and $l_{\mathcal{B}}^{\text{ENC}} + l_{\mathcal{B}}^{\text{KEY}} + l_{\mathcal{R}}^{\text{ENC}} + l_{\mathcal{R}}^{\text{KEY}}$ constraints.
2. For $\text{DoF}_{\mathcal{B}}$ values of $\blacksquare$, compute to $End_{\mathcal{B}}$ and store them in $L_1[End_{\mathcal{B}}]$.
3. For $\text{DoF}_{\mathcal{R}}$ values of $\blacksquare$, compute to $End_{\mathcal{R}}$ and store them in $L_2[End_{\mathcal{R}}]$.
4. Find match between L_1 and L_2.
5. For the surviving pairs that pass the match, check for a full-state match.

Framework of the Meet-in-the-Middle Attack

Attack Framework

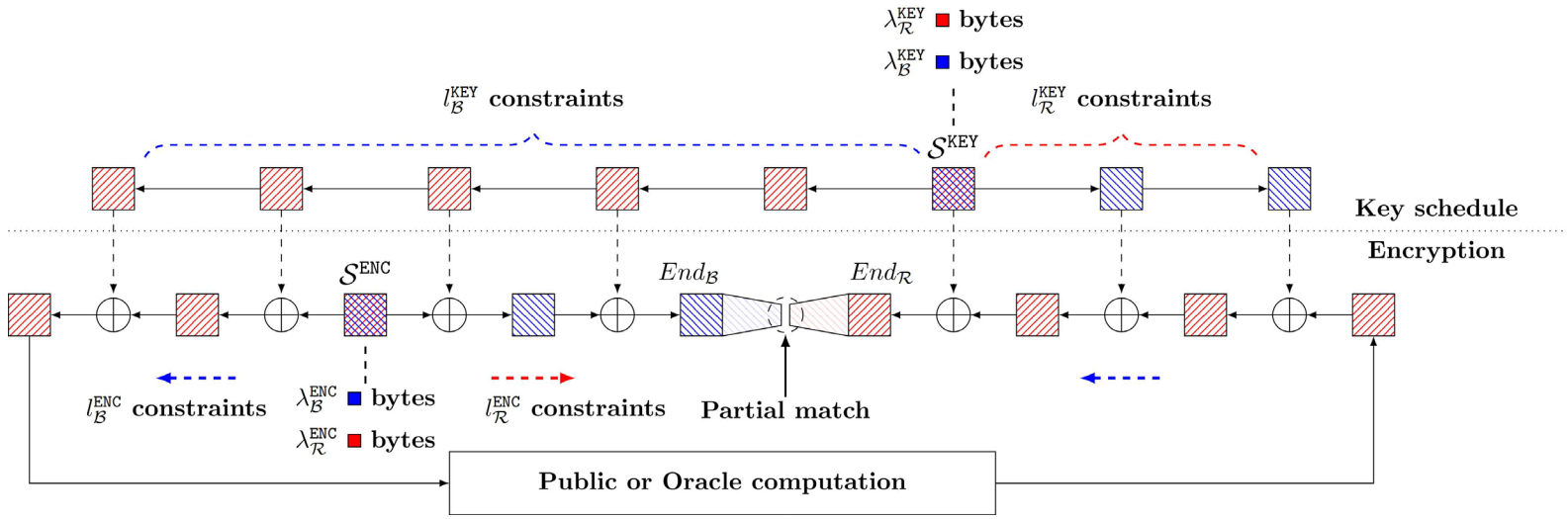
1. Choose constants in $\mathcal{S}^{\mathtt{ENC}}$ and $\mathcal{S}^{\mathtt{KEY}}$ and $l_{\mathcal{B}}^{\mathtt{ENC}} + l_{\mathcal{B}}^{\mathtt{KEY}} + l_{\mathcal{R}}^{\mathtt{ENC}} + l_{\mathcal{R}}^{\mathtt{KEY}}$ constraints.
2. For $\mathrm{DoF}_{\mathcal{B}}$ values of ■, compute to $End_{\mathcal{B}}$ and store them in $L_1[End_{\mathcal{B}}]$.
3. For $\mathrm{DoF}_{\mathcal{R}}$ values of ■, compute to $End_{\mathcal{R}}$ and store them in $L_2[End_{\mathcal{R}}]$.
4. Find match between L_1 and L_2.
5. For the surviving pairs that pass the match, check for a full-state match.

Cplx.

According to $\lambda_{\mathcal{G}} = n - \lambda_{\mathcal{B}} - \lambda_{\mathcal{R}}$, then $\lambda_{\mathcal{G}} + l_{\mathcal{B}} + l_{\mathcal{R}} = n - \mathrm{DoF}_{\mathcal{B}} - \mathrm{DoF}_{\mathcal{R}}$,

$$
\begin{aligned}
T &= 2^{8\times(n-\mathrm{DoF}_{\mathcal{B}}-\mathrm{DoF}_{\mathcal{R}})} \times \left(2^{8\times\mathrm{DoF}_{\mathcal{B}}} + 2^{8\times\mathrm{DoF}_{\mathcal{R}}} + 2^{8\times(\mathrm{DoF}_{\mathcal{B}}+\mathrm{DoF}_{\mathcal{R}}-\mathrm{DoM})}\right) \\
&= 2^{8\times(n-\min\{\mathrm{DoF}_{\mathcal{B}},\mathrm{DoF}_{\mathcal{R}},\mathrm{DoM}\})} \\
M &= \min\{2^{8\times\mathrm{DoF}_{\mathcal{B}}},\ 2^{8\times\mathrm{DoF}_{\mathcal{R}}}\}
\end{aligned}
$$

Framework of the Meet-in-the-Middle Attack[EC:BDG+21]



1. ■ $\mathcal{R}$, $(x, y) = (0, 1)$: Known byte only with backward computation.
2. ■ $\mathcal{B}$, $(x, y) = (1, 0)$: Known byte only with forward computation.
3. ■ $\mathcal{G}$, $(x, y) = (1, 1)$: Constant byte and known in both forward and backward computations.
4. □ $\mathcal{W}$, $(x, y) = (0, 0)$: Unknown byte in forward and backward computations.

Enhanced Techniques[C:DHSLWH21]

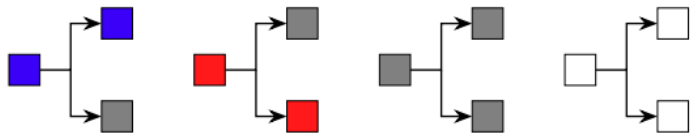
- Initial structure
  - Linear constraints: Solving linear systems of equations
  - Nonlinear constraints: Precomputing the concrete value of constraint constants $\mathfrak{c}_{\mathcal{B}}$ by traversing $\lambda_{\mathcal{B}}$ ■ and store $\mathrm{DoF}_{\mathcal{B}}$ ■ in table $U[\mathfrak{c}_{\mathcal{B}}]$. Similar to ■ and $V[\mathfrak{c}_{\mathcal{R}}]$.

Improved attack framework

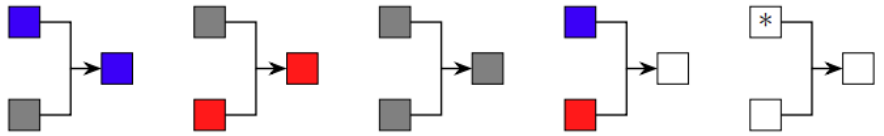
1. Choose constants for the starting states.
2. Build two table to solve nonlinear constraints $\mathfrak{c}_{\mathcal{B}}/\mathfrak{c}_{\mathcal{R}}$ with additional time $\max\{2^{8\times\lambda_{\mathcal{B}}},\ 2^{8\times\lambda_{\mathcal{R}}}\}$ and memory $\max\{2^{8\times\lambda_{\mathcal{B}}},\ 2^{8\times\lambda_{\mathcal{R}}}\}$.
3. For each $\mathfrak{c}_{\mathcal{B}}$, $\mathrm{DoF}_{\mathcal{B}}$ values of ■ are computed to $End_{\mathcal{B}}$ and store in $L_1[End_{\mathcal{B}}]$.
4. For each $\mathfrak{c}_{\mathcal{R}}$, $\mathrm{DoF}_{\mathcal{R}}$ values of ■ are computed to $End_{\mathcal{R}}$ and store in $L_2[End_{\mathcal{R}}]$.
5. Find match between L_1 and L_2.

Enhanced Techniques[C:BGST22]

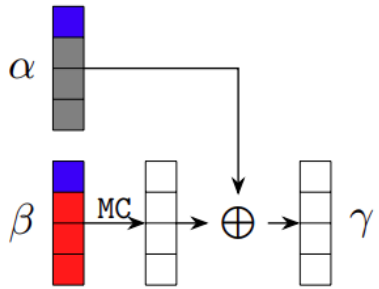
- Intermediate states are separated into two Superposition(SupP) states.
- SupP states are handled independently in linear operation and can be propagated equally in Bi-directions(BiDir).
- SupP states are combined before the next nonlinear operation.



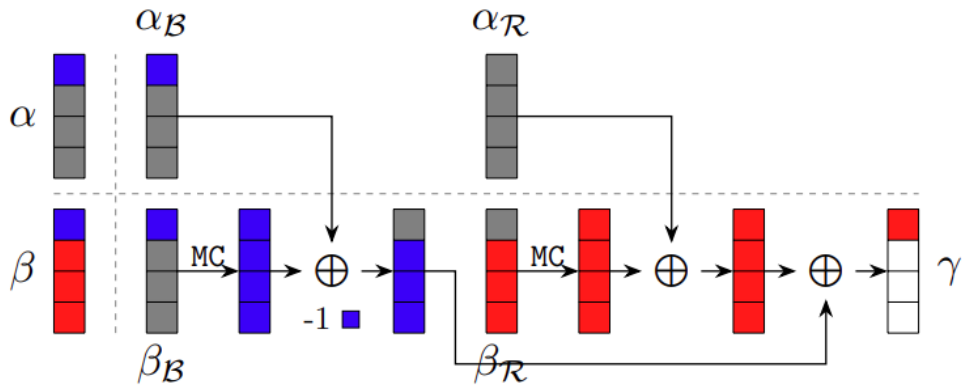
(a) Rules for separation



(b) Rules for combination



☒ MC and Xor

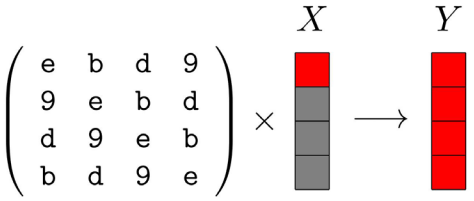


☑ SupP MC and Xor

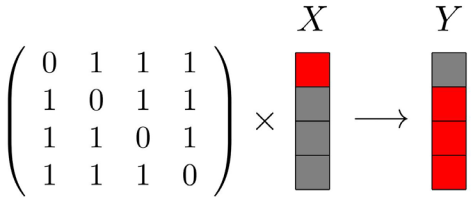
RSAC 2025 Conference

New models for linear layer with binary matrix

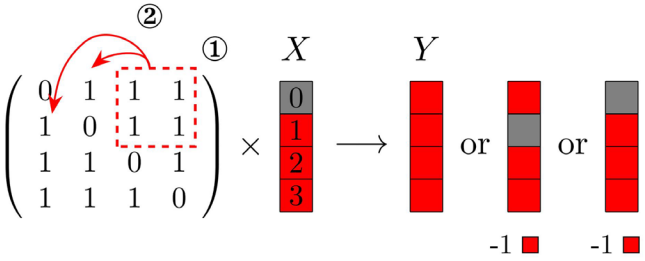
n-XOR Model



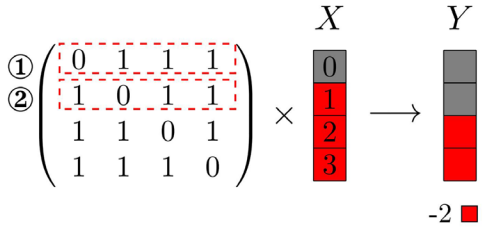
(a) Coloring pattern of `MC-RULE` for MDS matrix



(b) Coloring pattern for binary matrix



(a) Model of consecutive `XOR`



(b) `n-XOR` Model

n-XOR Model

- n-XOR-RULE^{-}-1. If there is at least one □ in input, then the output is □.
- n-XOR-RULE^{-}-2. If all cells of the input are ■, then the output must be ■.
- n-XOR-RULE^{-}-3. If there are ■ and ■ cells but no □ cell in the input, then one of the following situations will occur:
  - The output is ■ cell and no DoF is consumed.
  - The output is ■ by consuming one DoF of ■.

$$\begin{cases} \sum_{i=0}^{n-1} y_i^A + \mu \leq n \\ \sum_{i=0}^{n-1} y_i^A + n \cdot \mu \geq n \\ \sum_{i=0}^{n-1} x_i^A - \nu \leq n-1 \\ \sum_{i=0}^{n-1} x_i^A - n \cdot \nu \geq 0 \end{cases} \quad (2) \qquad \begin{cases} y^B + \mu = 1 \\ x^B + \mu \leq 1 \\ \eta - x^B + \nu = 0 \\ \sum_{i=0}^{n-1} x_i^A + x^B - 2 \cdot \nu \leq n-1 \\ \sum_{i=0}^{n-1} x_i^A + x^B - (n+1) \cdot \nu \geq 0 \end{cases} \quad (3)$$

Check Model: More Accurate Consumption of DoFs

$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \times \begin{matrix} X \\ 0 \\ 1 \\ 2 \\ 3 \end{matrix} \longrightarrow \begin{matrix} Y \\ \\ \\ \\ \\ \end{matrix}$$

i.e. n-XOR Model $\begin{cases} X[1] \oplus X[3] = c_0 \\ X[1] \oplus X[3] = c_1 \end{cases}$ -2 ■

Check Model $\Longrightarrow$ $\begin{cases} X[1] \oplus X[3] = c_0 \\ X[1] \oplus X[3] = c_0 \end{cases}$ -1 ■

(a) Possible situation I

$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \times \begin{matrix} X \\ 0 \\ 1 \\ 2 \\ 3 \end{matrix} \longrightarrow \begin{matrix} Y \\ \\ \\ \\ \\ \end{matrix}$$

i.e. n-XOR Model $\begin{cases} X[2] \oplus X[3] = c_0 \\ X[1] \oplus X[3] = c_1 \\ X[1] \oplus X[2] = c_2 \end{cases}$ -3 ■

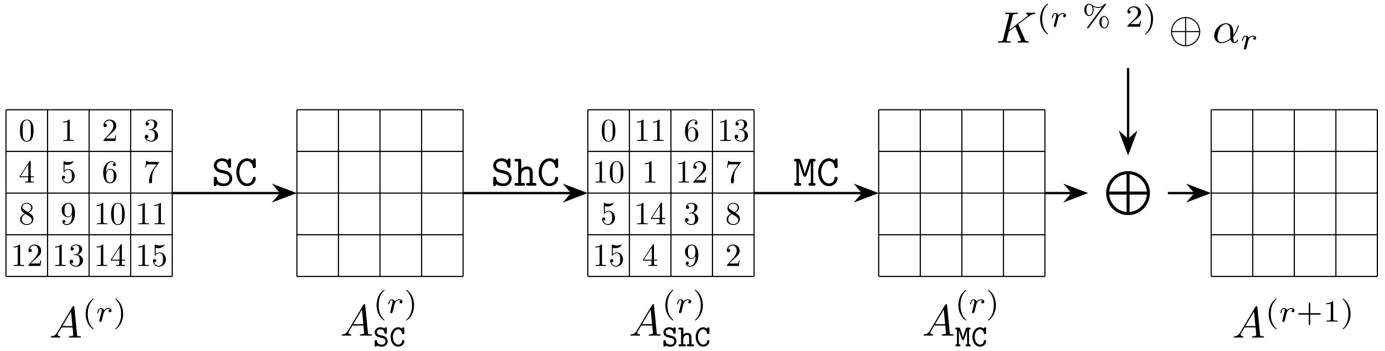
Check Model $\Longrightarrow$ $\begin{cases} X[2] \oplus X[3] = c_0 \\ X[1] \oplus X[3] = c_1 \\ X[1] \oplus X[2] = c_0 \oplus c_1 \end{cases}$ -2 ■

(b) Possible situation II

RSAC 2025 Conference

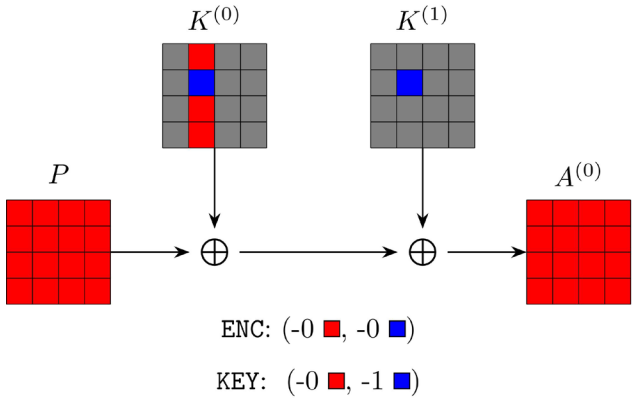
Key recovery attack on Midori64

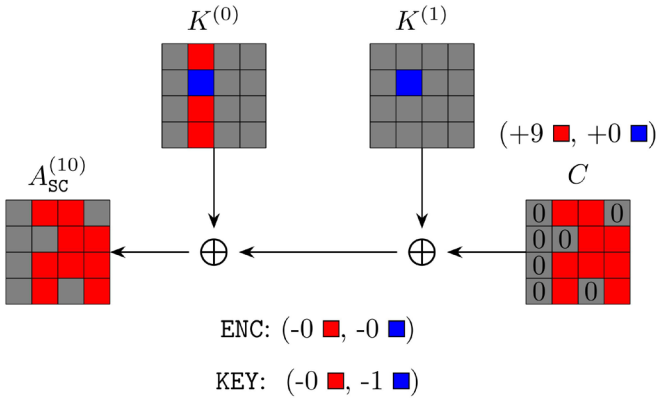
Key Recovery Attack on 11-round Midori64

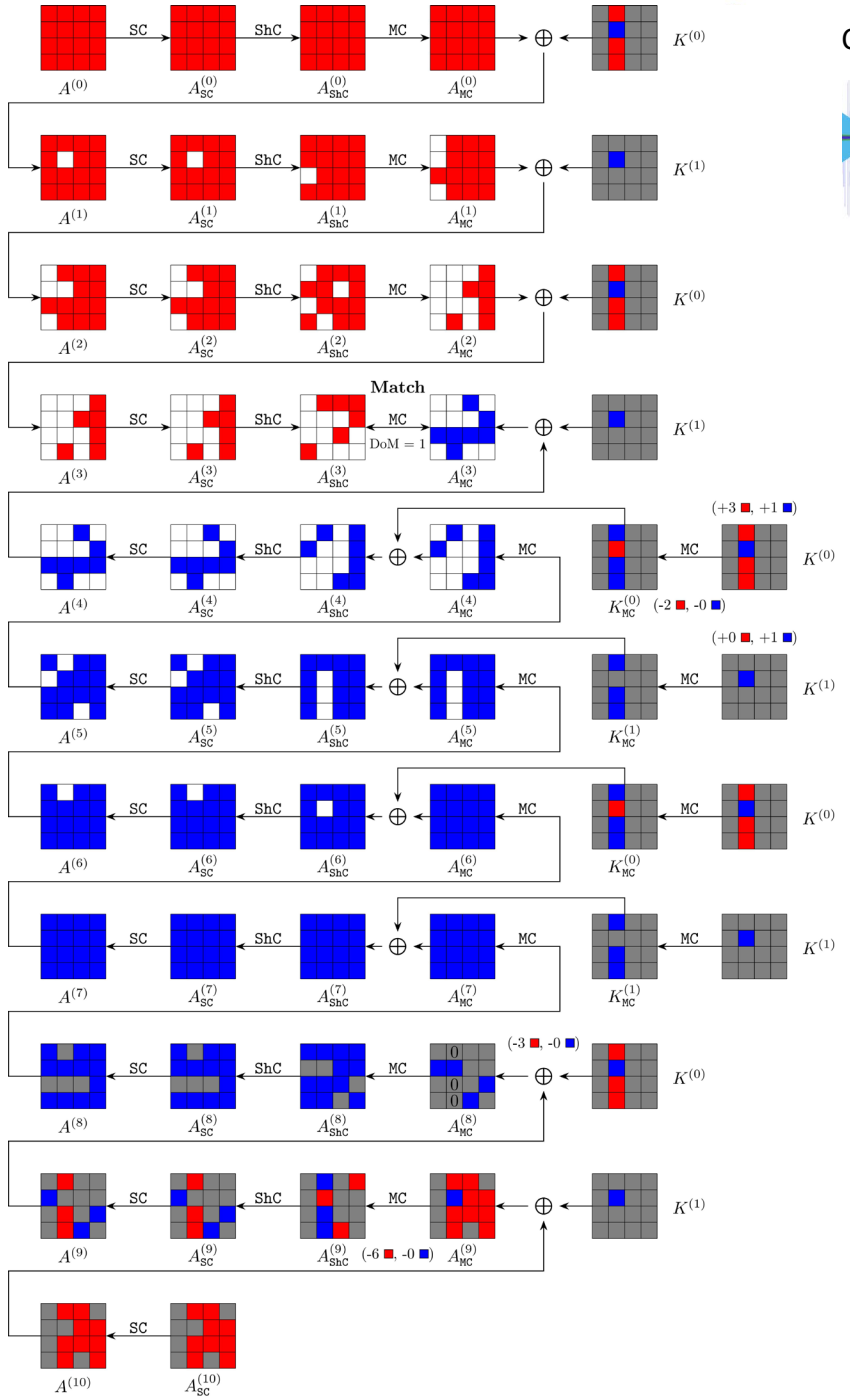


$$M = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}.$$

Key Recovery Attack on 11-round Midori64



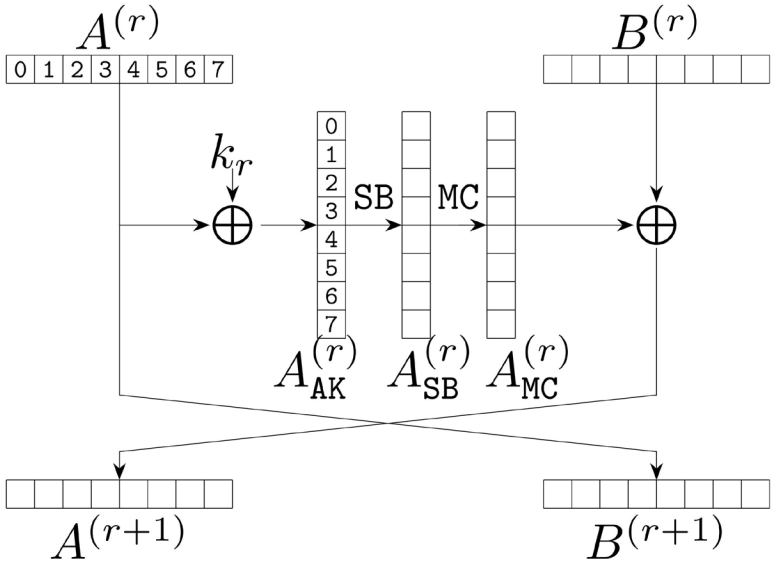


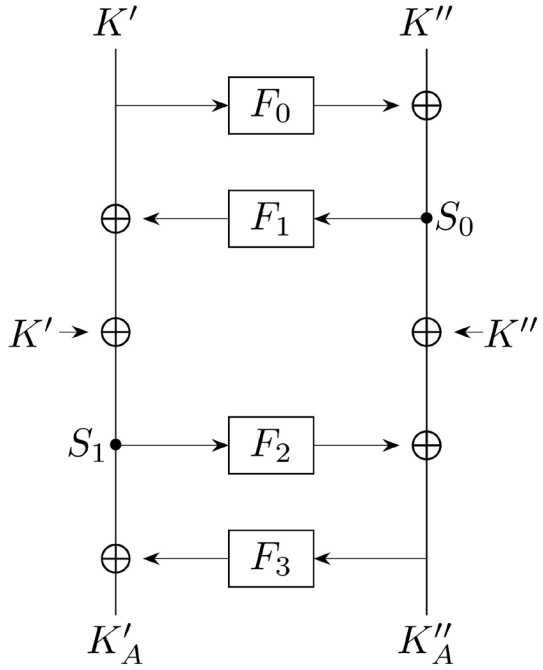


RSAC 2025 Conference

Preimage attack on weakened Camellia and Aria

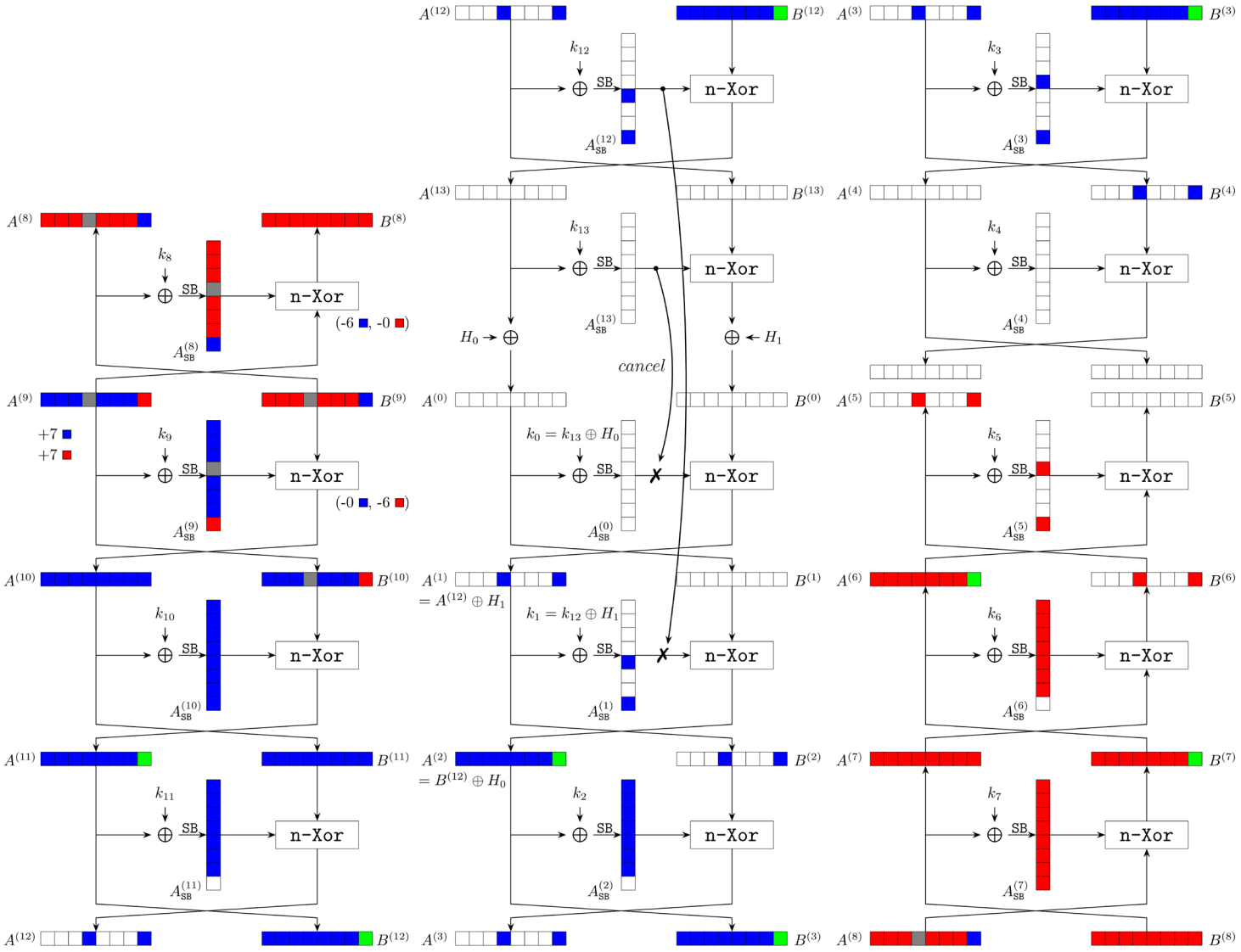
MitM Characteristic of 14-round weakened Camellia





$$P = \begin{pmatrix} 1\,0\,1\,1\,0\,1\,1\,1 \\ 1\,1\,0\,1\,1\,0\,1\,1 \\ 1\,1\,1\,0\,1\,1\,0\,1 \\ 0\,1\,1\,1\,1\,1\,1\,0 \\ 1\,1\,0\,0\,0\,1\,1\,1 \\ 0\,1\,1\,0\,1\,0\,1\,1 \\ 0\,0\,1\,1\,1\,1\,0\,1 \\ 1\,0\,0\,1\,1\,1\,1\,0 \end{pmatrix}.$$

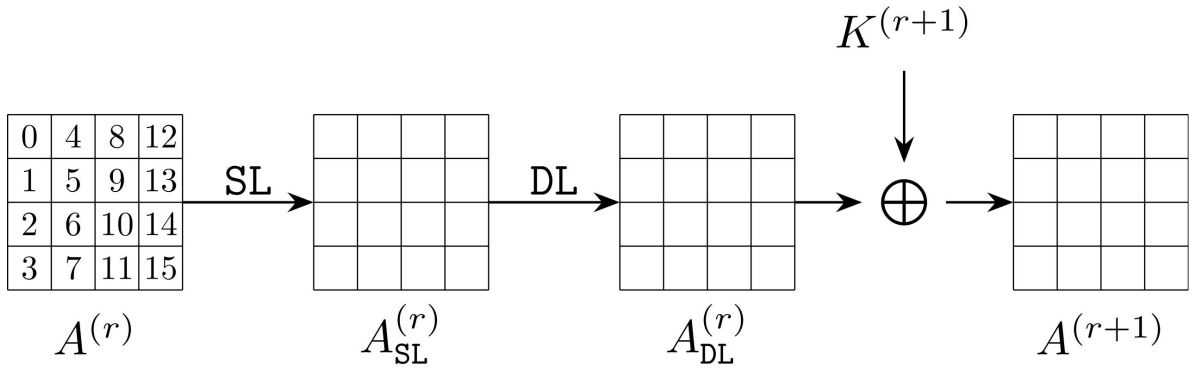
MitM Characteristic of 14-round weakened Camellia



(a) Initial structure (b) 4-round shrink and link (c) Matching point

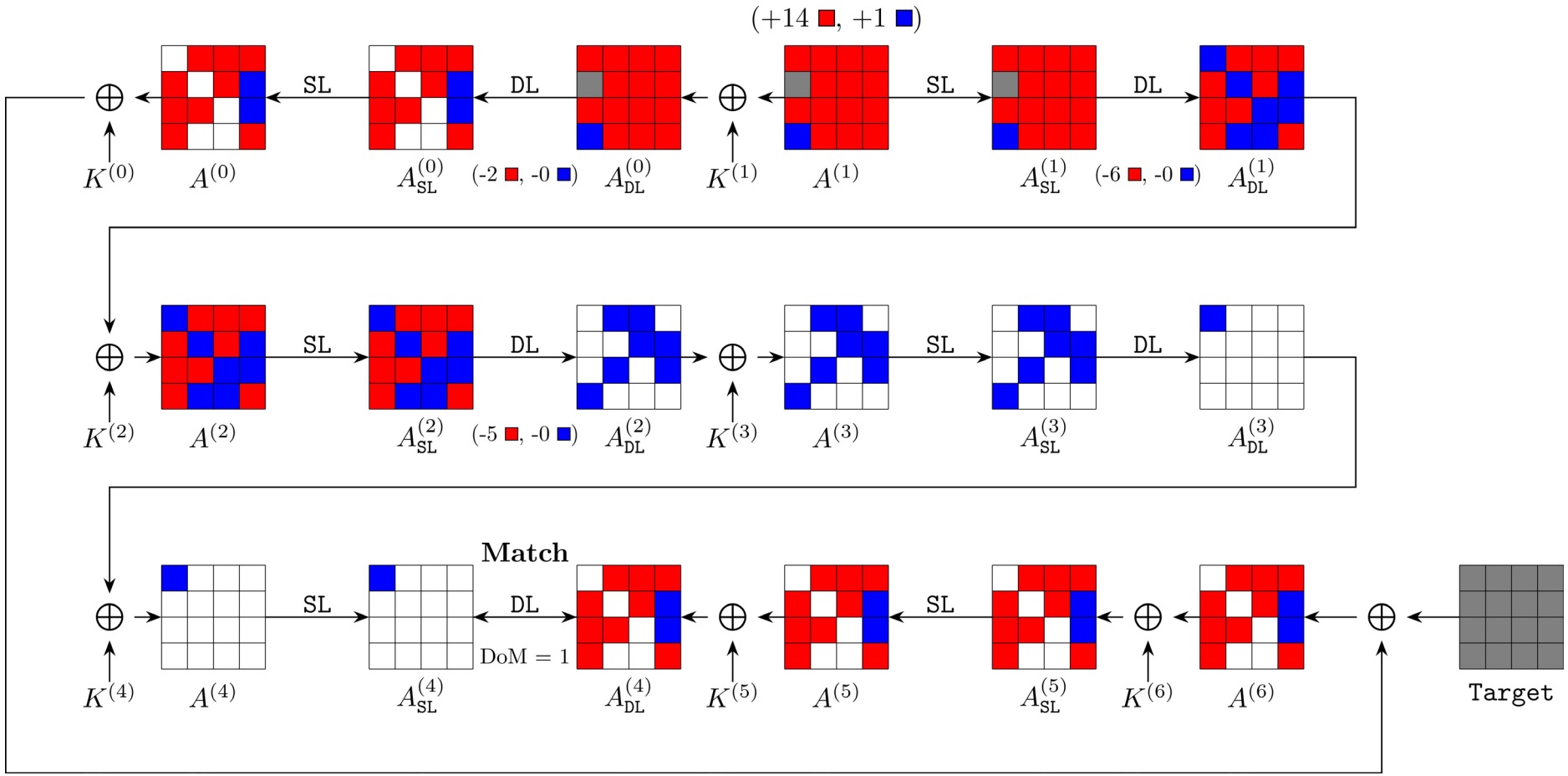
#RSAC
山东大学 SHANDONG UNIVERSITY
RSAC 2025 Conference

Preimage Attack on 6-Round Aria



$$P = \begin{pmatrix}
0&0&0&1&1&0&1&0&1&1&0&0&0&1&1&0\\
0&0&1&0&0&1&0&1&1&1&0&0&1&0&0&1\\
0&1&0&0&1&0&1&0&0&0&1&1&1&0&0&1\\
1&0&0&0&0&1&0&1&0&0&1&1&0&1&1&0\\
1&0&1&0&0&1&0&0&1&0&0&1&0&0&1&1\\
0&1&0&1&1&0&0&0&0&1&1&0&0&0&1&1\\
1&0&1&0&0&0&0&1&0&1&1&0&1&1&0&0\\
0&1&0&1&0&0&1&0&1&0&0&1&1&1&0&0\\
1&1&0&0&1&0&0&1&0&0&1&0&0&1&0&1\\
1&1&0&0&0&1&1&0&0&0&0&1&1&0&1&0\\
0&0&1&1&0&1&1&0&1&0&0&0&0&1&0&1\\
0&0&1&1&1&0&0&1&0&1&0&0&1&0&1&0\\
0&1&1&0&0&0&1&1&0&1&0&1&1&1&0&0\\
1&0&0&1&0&0&1&1&1&0&1&0&0&1&0&0\\
1&0&0&1&1&1&0&0&0&1&0&1&0&0&1&0\\
0&1&1&0&1&1&0&0&1&0&1&0&0&0&0&1
\end{pmatrix}.$$

Preimage Attack on 6-Round Aria



RSAC 2025 Conference

Conclusion

Conclusion

Table 1: Single Key attacks on `Midori64`, where ID and $\mathcal{DS}$-MitM denote impossiable differential and Demirci-Selçuk MitM attack, respectively.

Target	Rounds	Data	Memory(Bytes)	Time(Enc.)	Technique	Ref.
`Midori64`	11	2^{60}	$2^{95.8}$	$2^{116.6}$	ID	[23]
	11	2^{53}	$2^{92.2}$	2^{122}	$\mathcal{DS}$-MitM	[22]
	11	$\mathbf{2^{36}}$	$\mathbf{2^{6}}$	2^{124}	MitM	Section 4.1
	12	$2^{55.5}$	2^{109}	$2^{125.5}$	$\mathcal{DS}$-MitM	[22]
	$12^{\dagger}$	$2^{61.9}$	2^{44}	$2^{90.5}$	ID	[35]
	$12^{\dagger}$	$\mathbf{2^{48}}$	$\mathbf{2^{10.6}}$	2^{120}	MitM	Section 4.2
	$12^{\dagger}$	$\mathbf{2^{36}}$	$\mathbf{2^{5.6}}$	2^{124}	MitM	Section 4.2

† Weakened version without whitening layers.

Table 2: A Summary of the MitM Attacks on Hashing Modes.

Target	Attacks	Rounds	Time1	Time2	Memory	Technique	Ref.
`Camellia-MMO`	Preimage	$13^{\ddagger}$	2^{120}	2^{125}	2^{8}	MitM	[28]
		$14^{\ddagger}$	2^{120}	2^{125}	2^{8}	MitM	Section 5
`Aria-DM`	Preimage	5	2^{120}	2^{125}	2^{8}	MitM	[16]
		6	2^{120}	2^{125}	2^{112}	MitM	Section 6

- ‡ Weakened version without FL/FL^{-1} and whitening layers.
- Time1 represents the time complexity of pseudo-preimage. Time2 represents the time complexity of preimage attack converted from the pseudo-preimage attack according to [24, Fact9.99].

Conclusion

- Propose the n-XOR model to simulate the XOR operation with an arbitrary number of input cells.
- Introduce another check model to determine the exact DoFs consumption of MitM attributes propagation, to liminate the subtle inaccuracies caused by n-XOR model
- Leading to improved MitM attack on Midori, Camellia, Aria
- Limitations: the size of the check model is still limited by the number of input cells n and does not work well when n > 4 in this paper.

RSAC 2025 Conference

Many Voices.
One Community.

Thank You

Full Version: https://eprint.iacr.org/2025/082

Source Code: https://github.com/wenny-kt/MITM-Binary-Matrix